Uso di librerie preconfezionate

Cosa sono:

- pezzi di programma già scritto da altri; di solito ben collaudati;
- comprendono le definizioni delle funzioni e il codice eseguibile;
- per librerie di pubblico dominio è disponibile anche il sorgente;

Diponibili pubblicamente sono

- GNU scientific libreries (GSL)
- CERNLIB
- Netlib
- LAPACK (matrici)
- fftw (trasformate di Fourier)
- e molto altro ancora. . .

Cosa calcolano

- costanti fondamentali
- funzioni speciali
- polinomi
- radici
- matrici
- trasformate di Fourier
- etc. etc.

D. Perché non l'ha detto dall'inizio? Non avrei frequentato questo corso!
R. *Avete mai provato a far benzina se l'auto è elettrica? Anche chi non vuole diventare meccanico deve conoscere un poco le automobili. Questo corso è un'introduzione alla Fisica Computazionale, non un trattato!*

D. Io però voglio occuparmi di fisica e non di programmazione. Ho già progettato di scoprire il settimo quark
R. *Per gli esperimenti di fisica delle alte energie è necessario avere dei buoni programmi di simulazione Montecarlo. Sono di solito non-standard, e ve li dovete quindi scrivere da soli.*

D. Nel mio gruppo di ricerca fanno calcoli perturbativi all'ordine 3/2. Nel corso non ne ho sentito parlare
R. *Ora che conoscete i rudimenti del mestiere, potete studiare da soli le equazioni differenziali stocastiche o i problemi che più vi interessano*

Scrivere programmi con le gsl

- È necesario includere le definizioni delle funzioni utilizzate, che stanno in file .h forniti assieme alle librerie

- I file da includere possono essere uno o più. Di regola si include gsl.h, ma anche file più specifici per il problema che si deve risolvere, come gsl_integration.h

- linkare con opportune librerie -lgsl. il corrispondente file si chiamerà libgsl.la o libgsl.so (linking statico o dinamico). Anche qui può essere necessario linkare altre librerie

- *Importante*: leggere *molto attentamente* la documentazione

Esempi con le librerie gsl

- GNU Scientific Library (GNU's Not Unix);
- includono sempre gsl.h;
- link sempre -lgsl e spesso -lgslcblas;
- a seconda della particolare routine che si usa si devono includere altri file d'intestazione: ad esempio, se si usano generatori di numeri pseudocasuali può essere necessario includere anche gsl/gsl_rng.h mentre se faccio un integrale gsl/gsl_integration.h;
- i file d'intestazione definiscono i prototipi delle funzioni ma anche alcune costanti e molti tipi di dati.

Strutture dati

- Le librerie gsl definiscono alcune strutture dati per risolvere diversi problemi

- Un esempio tipico è gsl_vector, che è definito da

```
typedef struct
{
  size_t size;
  size_t stride;
  double * data;
  gsl_block * block;
  int owner;
} gsl_vector;
```

- Questa definizione è diversa da quella usata in questo corso

- Questo non vuol dire che le routine non possano essere usate, ma solo che occorre scrivere una funzione che permetta di utilizzarla in C++

Quando conviene usarle

Vantaggi Sono scritte da veri esperti della programmazione usando algoritmi sofisticati, capaci di gestire casi particolari e condizioni di errore difficilmente prevedibili. Il grande numero di utenti permette di sperimentarle e correggerle.

Svantaggi Il codice è meno portabile. La possibilità di usare lo stesso codice su di un altro computer dipende dalla disponibilità delle stesse librerie.

Quando conviene usarle Per problemi molto delicati e complessi come tutti quelli legati alle matrici, la generazione di numeri casuali, la ricerca di zeri e minimi di funzioni; oppure quando la velocità è un fattore critico, spesso nelle FFT.

Esempi

- costanti matematiche
  M_PI $= \pi$

- funzioni elementari e funzioni speciali
  hypot(3, 4)
  gsl_pow_4(2) potenza
  gsl_sf_bessel_J0 funzione di Bessel cilindrica

- valutazione di polinomi
  $p(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$
  gsl_poly_eval(c, n, x)

- ordinamento di un vettore
  gsl_sort(v, k, n)
  v(0), v(k), v(2k), ...
  per un vettore di n elementi

In pratica

Includo i file header

#include < gsl/gsl_xxx.h >

xxx = math, poly, sort_vector

Linking

gcc -lgsl -lgslcblas programma.c (-static)

BLAS = Basic Linear Algebra Subroutines

Documentazione

Mi documento con info gsl guardando soprattutto gli esempi.

Tre esempi

1. Integrazione

- Il file da includere è ⟨gsl/gsl_integration.h⟩
- è necessario fornire dello spazio "extra" di lavoro usato dalla routine
- la funzione da integrare è parte di una struttura
- la funzione può dipendere da altri parametri oltre alla variabile su cui integro. Devo passare questa informazione alla funzione in qualche modo: in pratica si mettono tutti i parametri in una struttura e si passa un puntatore ad essa.

2. Sorting

- Ordina un array
- Utilizza una struttura particolare del vettore
- Occorre includere il file ⟨gsl/gsl_sort_double.h⟩ e tenere conto della diversa struttura dei vettori

3. Generazione di numeri pseudocasuali

- Il file da includere è ⟨gsl/gsl_rng.h⟩.

- il programma fornisce una funzione che ritorna un numero pseudorandom tra zero e uno. Il metodo da usare è determinato dall'utente che deve fornire il tipo con la chiamata ad una opportuna funzione di inizializzazione.

- come per i numeri Fibonacci lagged, e' necessario avere a disposizione della memoria extra. Un'apposita funzione si occupa di allocarla: chiaramente la quantità di memoria allocata dipenderà dal tipo di generatore usato.